A Multimedia Streaming System for Urban Rail Environments

Justas Poderys[1(✉)], Jahanzeb Farooq[1,2], and Jose Soler[1]

[1] Technical University of Denmark, Kongens Lyngby, Denmark
{juspo,jafar,joss}@fotonik.dtu.dk
[2] Siemens A/S, Ballerup, Denmark
jahanzeb.farooq@siemens.com

Abstract. Due to a large number of mostly stationary users inside a train and the availability of a radio connection to the outside world, urban rail environments serve as promising candidates for multimedia distribution systems deployment. This work proposes to offload the individual per-passenger cellular network connections by using the excessive Communications-Based Train Control (CBTC) radio link bandwidth to deliver multimedia streams to a train, where it is subsequently distributed to the passengers using peer-to-peer based data distribution. Connections among the train passengers are implemented using the Wi-Fi Direct connectivity and data exchange is coordinated by using the Peer-to-Peer Streaming Peer Protocol. This work presents the solution and evaluates it in the scope of urban rail deployment. Network emulation tests are used to analyze the factors impacting the number of concurrent users that can use the proposed system. This work also proposes future work lines that can be used to improve the system's design.

Keywords: Wi-Fi direct · Peer-to-Peer Streaming Peer Protocol · Train communication · Streaming multimedia · Peer-to-peer communication

1 Introduction

The amount of data transferred by mobile devices is growing at a high rate. According to the Ericsson Mobility Report [1], data usage by smartphone users is growing at 40% compounded annual growth rate (CAGR). The CAGR of video streaming data to mobile devices is 62% and that of audio streaming is 45% according to the Virtual Networking Index report [2] by Cisco. Between 30 and 40% of users stream video and audio data during their commute [1].

At present, urban rail passengers receive mobile data from cellular networks (see panel A of Fig. 1). During high popularity events (e.g. sports matches, live shows, etc.) cellular network can get overwhelmed by user requests to deliver multimedia data. Urban rail environments serve as a promising candidate for deployment of multimedia distribution systems that can be used to offload cellular networks. As urban rail contains a large number of mostly stationary users in close proximity, it allows for usage of wireless peer-to-peer communication technologies for multimedia data distribution.

Furthermore, the availability of radio communication technologies in modern urban rail environments allows for the delivery of bandwidth-intensive data to the train, where it is subsequently distributed to the passengers.
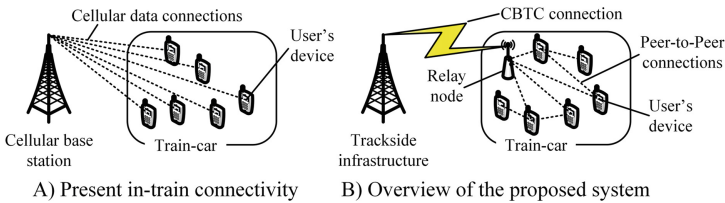
Fig. 1.   Present and proposed methods of streaming multimedia distribution to the train passengers

Conventional railway signaling systems, based on color light signals, are rapidly being replaced by modern, electronic systems, widely known as communication-based signaling systems [3, 4]. One of the most notable of such systems is Communications-Based Train Control (CBTC) [5]. In communication-based signaling, a radio communication technology is used to transfer trains control information between a train and the train control center. Nearly all of the modern CBTC systems use IEEE 802.11, widely known as Wi-Fi [6], as the radio technology, mainly owing to its cost-effectiveness [7]. IEEE 802.11a or 802.11g are the most commonly used versions of this technology, supporting data rates of up to 54 Mbps. More advanced versions such as 802.11n and 802.11ac can support even higher rates, up to 1 Gbps.

Using radio communication, CBTC systems enable transfer of real-time and high-resolution train control information between a train and the train control center. CBTC data, which consists of train control messages typically sent about every 500 ms, itself does not require high bandwidth. Typical data rate ranges up to maximum 100 kbps [7]. This leaves with plenty of excessive bandwidth to be used for other purposes.

This work proposes an in-train data distribution system that can be used to distribute streaming multimedia to the train passengers using the unused CBTC radio link bandwidth (see panel B of Fig. 1). Once the multimedia stream is delivered to the train over the CBTC radio connection, it is distributed to the train passenger devices by using peer-to-peer technologies (see Sect. 3 for a detailed description): data connections between the passenger devices are implemented using Wi-Fi Direct [8] and the multimedia data exchange is coordinated by using the Peer-to-Peer Streaming Peer Protocol [9]. While all types of data can be delivered using P2P communication paradigm, this work focuses on streaming multimedia, as it is the biggest driver of mobile data usage.

The goal of this work is to evaluate the proposed in-train multimedia delivery solution. The evaluation is performed by emulating the system, to establish the highest number of concurrent users that can receive the same live High-Definition (HD 720p) multimedia stream. Furthermore, during the system tests, three parameters impacting the streaming multimedia user's experience (see Sect. 5 for a detailed description) were observed: the playback index, the amount of video data skipped and the time until the first frame is rendered.

This work considers streaming multimedia delivery to a single train-car. A train-set with multiple cars could have the proposed system installed in each train-car. While this work considers data distribution performance, it leaves data security and integrity issues for future research.

2   Related Research

The prior research related to the proposed multimedia distribution is grouped into two groups. The first group lists research related to the connectivity methods and deployment scenarios. The second group contains an overview of the group communication protocols.

Research in the first group considers wireless multicast communication. Multicast is a one-to-many communication paradigm, well suited to implement data distribution to a group of users. The authors of [10] propose to deliver multimedia content to the Wi-Fi users via the system called DirCast. Their system uses infrastructure (Access Point) based Wi-Fi connections and requires that the clients' wireless adapters are set to the promiscuous mode. The authors tested the DirCast system with up to 12 users and concluded that DirCast is well suited for streaming media delivery. The authors of [11] investigate the applicability of the IEEE 802.11b/g/n technologies for multicast-based multimedia streaming. The evaluations are based on the mean opinion score of the evaluators. Test results showed that connections using 802.11 g and 802.11n technologies can be used for multicast-based multimedia delivery. However, the achieved data rate differed between Wi-Fi adapters from different manufacturers, even if the adapters supported the same Wi-Fi standard. The authors of [12] consider Wi-Fi based multicast deployment in high-speed trains. In contrast to the work presented here, they consider WiMAX technology to deliver the media stream to the train. The early results indicate that while their approach is feasible, it suffers from a high packet loss.

Research considering Wi-Fi Direct as a communication technology is limited, compared to research considering Wi-Fi based multicast. Authors of [13] presented an in-depth Wi-Fi Direct technology overview and conducted experiments using Wi-Fi Direct to offload 3G (HSDPA) based connections between two computers. Their results show that Wi-Fi Direct is a viable solution for short-range device-to-device communication, achieving data rates up to 3 times higher than using a 3G cellular connection. Research presented in [14] use analytical and simulation analysis to show that Wi-Fi Direct can be used to offload LTE-based cellular connections between the devices in close proximity.

The second group of related work considers application layer multicast (ALM) protocols that can be used to implement one-to-many group communication. Authors of [15] survey ALM protocols specifically designed for live peer-to-peer video streaming. The multimedia distribution system presented in this work uses one of the surveyed protocols: Peer-to-Peer Streaming Peer Protocol [9] (PPSPP). PPSPP is based on the Swift protocol. The authors of [16] provide an in-depth analysis of the Swift protocol features, while the authors of [17] provide a performance analysis of the Swift protocol. The PPSPP protocol specification allows the protocol implementer to choose
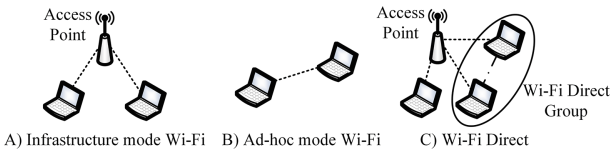
any data transmission scheduling algorithm. Authors of [18] provide examples of several of such algorithms along with their performance evaluation.

3   Architecture Overview

The proposed in-train multimedia distribution system is implemented using two core technologies to deliver multimedia data to the train passengers. Data connections between the passengers are implemented using the Wi-Fi Direct technology and the multimedia sharing is coordinated using the Peer-to-Peer Streaming Peer Protocol. This section describes the details of both technologies and then presents the overall architecture of the in-train multimedia distribution system.

3.1   Wi-Fi Direct

In the conventional, infrastructure mode of IEEE 802.11 Wi-Fi, two devices placed side by side must connect to an Access Point (AP) in order to be able to communicate to each other, as shown in panel A of Fig. 2. The AP, in this case, is often a dedicated hardware with a superior set of capabilities. An AP is responsible for implementing functionalities such as security, power saving and QoS in the Basic Service Set (BSS), and connects the BSS to the infrastructure network.

Access
Point   Access
                             Point   Wi-Fi Direct
                                                        Group

A) Infrastructure mode Wi-Fi   B) Ad-hoc mode Wi-Fi   C) Wi-Fi Direct

Fig. 2. Different modes of Wi-Fi deployment

Users like to share content and services on their personal devices as they interact with other users on the go, without the need for first connecting to an AP [19]. Hence the need for device-to-device (D2D) communication in today's highly dynamic world is rapidly growing. While the 802.11 Wi-Fi has been around for almost two decades now, the D2D communication it supports with its infrastructure-less mode (also known as "ad-hoc mode", shown in panel B of Fig. 2) is far from trivial to use [19]. Furthermore, since infrastructure-less mode does not involve an AP, it lacks the above-mentioned functionalities normally implemented by an AP.

In Wi-Fi Direct, communication takes places in the form of groups. In a P2P Group, a Wi-Fi Direct device—referred to as a P2P Device—can have the role of a P2P Group Owner (GO) or a P2P Client. These roles are negotiated when the group is formed. A GO—also referred to as a "Soft AP" occasionally—provides AP-like functionalities to its clients, both legacy (i.e. conventional 802.11 devices) and P2P clients [19]. GO is as well responsible for running a DHCP server to assign IP addresses to the clients. Like in a conventional Wi-Fi network, a P2P Group has an SSID. This SSID always starts with ASCII characters "Direct-" to distinguish itself from conventional Wi-Fi SSIDs [8].

One may argue that Wi-Fi Direct is not a peer-to-peer technology in its true sense as there exist hierarchical roles (GO and clients). This is a contrast to the conventional ad-hoc networking where all devices are equal, without any hierarchies.

A Wi-Fi device must support IEEE 802.11 g or a newer technology to be able to act as P2P Device [8]. A P2P Device can assume both roles (GO and Client) if it has multiple wireless interfaces, or if it can implement virtual interfaces or time-sharing techniques on the same interface [19]. Likewise, a P2P Device can be both a part of a Wi-Fi Direct group and maintain a conventional Wi-Fi connection simultaneously, as shown in panel C of Fig. 2. Such a device is referred to as a P2P Concurrent Device [8, 13, 19]. This enables a wide range of possibilities. For example, a P2P Device can provide a connection to the outer world by simultaneously connecting to a conventional Wi-Fi AP or a 3G/4G network and to a P2P Client. This is a major advantage over conventional ad-hoc networking in which devices cannot simultaneously connect to existing Wi-Fi networks. Additionally, a P2P Device can be a member of multiple P2P Groups simultaneously [8].

Thus, in contrast to a conventional Wi-Fi network in which a device can either operate as an AP or a client, in Wi-Fi Direct, these roles are arbitrary—any device can function as an AP. Furthermore, the fact that this new capability can be implemented entirely in software, without a need of making any changes to the existing 802.11 hardware, bears a high significance [13].

3.2   Peer-to-Peer Streaming Peer Protocol

Streaming multimedia distribution to and between the train passengers is coordinated by using the Peer-to-Peer Streaming Peer Protocol [9] (PPSPP). PPSPP is an Internet Engineering Task Force standardized protocol used to implement peer-to-peer based data distribution.

Each train passenger using the in-train streaming multimedia distribution system has to run a client of PPSPP protocol software. The client software can be run on any computing device, like a laptop, a smartphone or a tablet. A group of connected users that are running the PPSPP clients is referred to as a swarm. After starting the PPSPP client, each member of a swarm makes a TCP connection to the tracker node. The tracker node is used by the arriving nodes to learn IP addresses of other members in the swarm and to get the bootstrap information required to start downloading data. The tracker node functionality can be implemented in a dedicated node or in one of the swarm members.

PPSPP works by dividing the shared data into fixed size pieces called chunks. Each client participating in the peer-to-peer data exchange periodically informs other clients about data chunks it has. It does so by sending one or more "HAVE" messages to all other connected clients. Each "HAVE" message contains a continuous range of chunks numbers that the client shares. Upon reception of the "HAVE" message, each client compares the range of chunks advertised in the "HAVE" message to the list of chunks available locally by the receiving client. If the sending client has chunks that are not available at the receiving client, the receiving client sends a "REQUEST" message to the sender of the "HAVE" messages, requesting the range of chunks not available at the

receiving client. Upon receiving a "REQUEST" message, a client starts sending the requested data chunks to the sender of the "REQUEST" message.

In the proposed system, multimedia data is 'injected' into the PPSPP swarm by a special node called the root node. Functionally, the root node is identical to the other nodes in the swarm running an instance of the PPSPP protocol. The only difference is that in the proposed system, tracker functionality was implemented in the same node as root node.

Clients in a swarm can use either TCP or UDP to communicate and exchange data chunks. When PPSPP is used with TCP protocol, repeating failed chunk transmissions is handled by the TCP protocol. Using UDP with PPSPP requires the implementer to design a custom automatic repeat request (ARQ) algorithm to reschedule failed chunk transmissions. To streamline the operation of the PPSPP clients, the in-train multimedia distribution system uses TCP protocol. When PPSPP is used to distribute streaming multimedia, each client can be configured with a discard window, indicating the number of data chunks that each client will hold and make available to other nodes before discarding them. Using a discard window reduces the memory requirements in the PPSPP clients, as each client keeps no more chunks than the size of the discard window.

3.3 Multimedia Distribution System Architecture

The overall multimedia distribution system architecture consists of a single radio connection to the train-car, the root node, and the in-train Wi-Fi Direct network, as shown in Fig. 3.
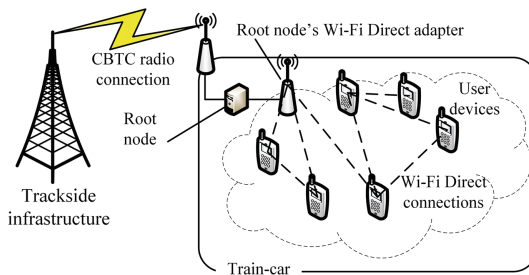
Fig. 3. Architecture of the proposed in-train multimedia streaming system

The multimedia stream is delivered to the train-car by utilizing the unused bandwidth of the CBTC radio connection. Nevertheless, the exact technology of the connection (Wi-Fi, LTE, IEEE802.11p, etc.) does not impact the functioning of the multimedia delivery system, as long as the connection can maintain the required data rate.

Once the multimedia stream is delivered to the train-car, it is received by the root node. The root node acts as a proxy server, receiving the multimedia stream and sharing it using the PPSPP protocol. The root node can be implemented using a commodity computer hardware. The requirements for the root node hardware are that it must support the same radio technology as used by the CBTC connection, must be able to interface

with the Wi-Fi network card supporting the Wi-Fi Direct protocol, and must be able to run an instance of the client software implementing the PPSPP protocol.

The train passengers can join the Wi-Fi Direct network by using a device supporting the Wi-Fi Direct connectivity (supported in systems running Android 4.0+ and MS Windows 8+) and having a client application installed on their devices. The application can be configured to join the local Wi-Fi Direct network by scanning a QR code located in the train-car or by using other location-based configuration methods. In addition to configuring the device's Wi-Fi Direct connectivity, the application acts as a client of the PPSPP protocol.

Once the train passenger's device is configured to join the Wi-Fi Direct network, it registers with the tracker node and starts making connections to other users. As soon as the first Wi-Fi Direct connection is established (either to the root node or to another user), the passenger starts receiving video data which is rendered on the device's screen. The proposed system has a single tracker node, co-located with the root node. For simplicity, other nodes in the system (passenger devices) never perform tracker node's functions.

4 Experimental Setup

To evaluate the proposed in-train streaming multimedia distribution system, a series of system emulations were performed. This section describes the experimental setup.

The multimedia distribution system was evaluated by using the CORE network emulator [20] with Extendable Mobile Ad-hoc Network Emulator (EMANE) [21]. The CORE network emulator uses Linux Containers (LXC) to divide the host computer running the CORE software into a number of independent virtual hosts, each running the Linux operating system. In our experiments, a virtual host represented a single train passenger using the multimedia distribution system. Each virtual train passenger ran an instance of the PPSPP protocol client[1]. Network emulation software ran in a server-grade workstation to reduce any negative impact of the availability of computational resources on emulation results. Each emulation scenario was repeated 3 times and the results were averaged.

The EMANE emulator was used to emulate the media access (MAC) and physical (PHY) layers of the Wi-Fi Direct protocol. The MAC layer used the IEEE 802.11g standard with the maximum data rate set to 54 Mbps. The Wi-Fi Direct connections were modeled by using the infrastructure-less Wi-Fi mode. In the current setup, all nodes use the same Wi-Fi channel to transmit data. In future work, multiple non-overlapping Wi-Fi radio channels will be used for communication between the nodes. This approach should minimize radio interference among the users and allow for a larger number of concurrent users.

Testing of the system used a model data source having the same data production rate as the VP8 video codec [22]. VP8 is a royalty-free video codec developed by Google and released to the public domain. The data rate of the codec was 2 Mbps and the video

---

[1] Source code available at http://github.com/justas-/PyPPSPP.

size of 1280 × 720 pixels (also known as HD 720p). The frame rate was set to 10 frames per second. The average key frame size was 29 301 B, and the average size of a non-key frame was 6 030 B. Audio data was encoded using "Vorbis" codec with an average frame size of 1 601 B.

5   Results Analysis and Evaluation

The proposed multimedia distribution solution was evaluated in a series of deployment scenarios using the CORE network emulator software. Each scenario varied by the number of users (between 14 and 20) and the number of concurrent connections each user maintained to other users (between 4 and 10). The following section describes the analyzed parameters and presents the experimental results. The analyzed parameters impacting user's experience were the playback index, the amount of skipped data and the time until the first frame is rendered.

5.1   Playback Index

In video streaming systems, video data is delivered to the clients divided into frames. The frames are then rendered on the device's screen by the video decoder at a constant frame rate. The playback index is the ratio of the number of frames rendered successfully to the number of render attempts. If the video decoder always has enough data in the buffer to render the next frame, the number of successful attempts is equal to the render attempts and the playback index is 1 (or 100%). From the streaming video user's perspective, a playback index of 100% corresponds to a video stream that is rendered without interruptions. The observed playback index in different emulated scenarios is shown in Fig. 4.
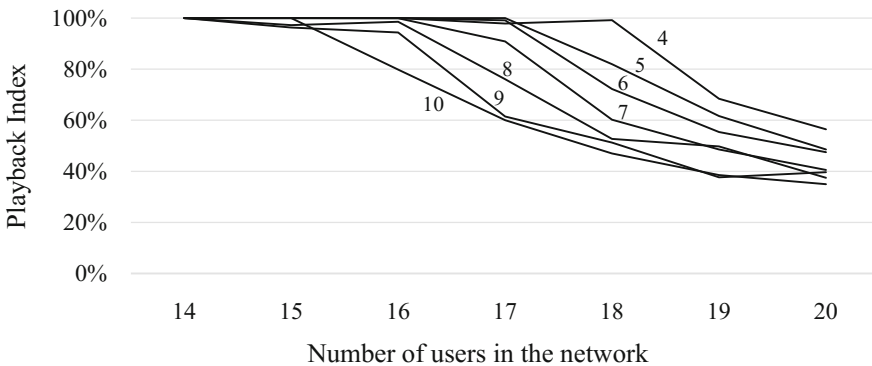
Fig. 4. Average playback index observed in test scenarios. Each line indicates the number of concurrent connections between users

The playback index is directly related to the availability of video data that is to be rendered. As Fig. 4 shows, the playback index (and the availability of video data) is

linked to the number of users in the network and the number of concurrent connections each user maintains to other users. As shown in the figure, the proposed system can support up to 15 concurrent users, while maintaining the playback index at 100%. The number of concurrent users can be increased to 18 without affecting the playback index by reducing the number of parallel connection each user maintains from 10 to 4.

The number of concurrent users is practically limited due to the users contending for the shared transmission medium. Each user actively transmitting data over the Wi-Fi Direct connection acts as a source of interference to the other users. Increasing the number of concurrent users increases the total amount of interference, which in turn reduces the rate of successful data transmissions between the users. This, in turn, reduces the availability of video data to the video decoder and reduces the playback index.

The number of concurrent connections a user maintains to other users has a similar effect on the playback index as the number of users in the network. Each concurrent connection is used to periodically exchange chunks availability information between the users. As the number of concurrent connections increases, each user transmits data more often, thus interfering with transmissions of the other users.

The number of concurrent users in the network can be increased (without reducing the playback index) by limiting the interference each user creates to other users. As mentioned above, in future, the user model will be upgraded to establish connections using multiple non-overlapping Wi-Fi channels to limit interference.

5.2   Skipped Data

All users in the proposed multimedia delivery system maintained a discard window of 1000 chunks. As described in Sect. 3.2, maintaining a discard window reduces the user's memory requirements. Since all users maintain the discard window individually, it is possible for the user's video decoder to get stuck by waiting for the frame that is already discarded by all remaining users in the swarm. In such case, user's video decoder has been programmed in a way that after 10th unsuccessful attempts to render a frame (video freeze of 1 s), it skips the following 250 chunks (the size of playback buffer) and continues rendering the video data. From the viewer's perspective, skipped data is seen as a jump forward in the playback of the video. The average chunk production rate of the video source was 80 chunks per second, hence each jump of 250 chunks corresponded to skipping of approximately 3,125 s of video data. The average number of skipped data chunks in the experiments is shown in Fig. 5.

The skipping of data chunks is caused by the video decoder being unable to decode the next frame in 10 consecutive attempts, due to the unavailability of video data. As such, the observed results follow closely the results of the playback index discussed in the previous section. The key cause of data being unavailable is users' contention for shared transmission medium and subsequent unsuccessful data transmissions.

As seen in Fig. 5, the proposed multimedia distribution system can deliver video data without skipping any data chunks to 14 users. The number of concurrent users that received video data and does not skip any data chunks can be further increased to 18 by reducing the number of connections that each user maintains to other users.
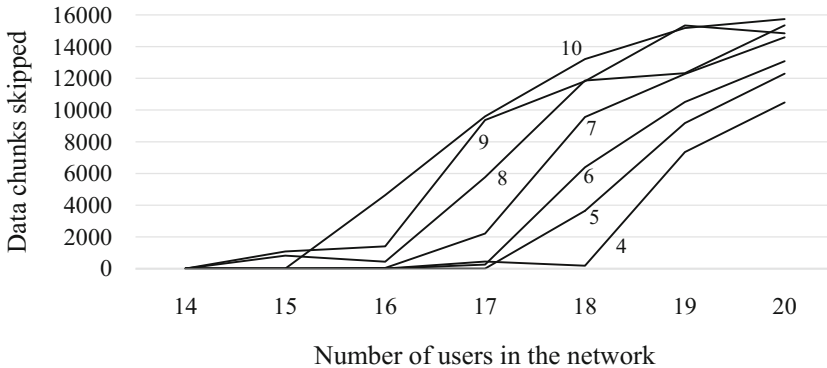
Fig. 5. Average number of skipped data chunks. Each line indicates the number of concurrent connections between users

The number of discarded data chunks can be decreased by reducing the interference in Wi-Fi Direct data transmissions between the users (as explained in the previous section) and by employing more advanced chunk skipping algorithms. In the current chunk skipping algorithm, users skip 250 chunks after a 10th consecutive unsuccessful attempt to render a video frame. A future version of the chunks skipping algorithm would skip missing chunks until a continuous block of data chunks of a certain size is found. Likewise, at the moment, if the unavailable chunks are delivered after they have been skipped, they are discarded. In future, a feature of cancelling the request for the chunks that have already been skipped will be considered.

5.3   Time to the First Frame

The last analyzed parameter is the length of the time period between the time a user's device starts joining the multimedia distribution system and the time the device renders the first frame. During this time period, the PPSPP client on the user's device is initialized, registers with the tracker node, learns about available data, connects to other devices (either other users or the root node), and receives enough data to fill the initial playback buffer (250 data chunks). The observed time to the first frame is shown in Fig. 6. For brevity, only the results for 4 to 8 concurrent connections are shown.

As seen in Fig. 6, the time period until the first frame is rendered increases with the size of the network. The average time to the first frame in the network of 14 users is 12.5 s, while in the network of 20 users it increases to 17.9 s. The average time to the first frame varies only slightly in the scenarios where each user maintains 6 or more concurrent connections to other users. On the other hand, it varies greatly in the scenarios where each user maintains only 4 or 5 concurrent connections to other users.

Overall the time to the first frame is generally long (between 12 and 17 s). The main cause of the long delay in the scenarios with 4 or 5 concurrent connections is the time required to find a node that can accept a connection. When each node is configured to maintain only a small number of connections, the probability of finding a node that can accept a new connection is lower.
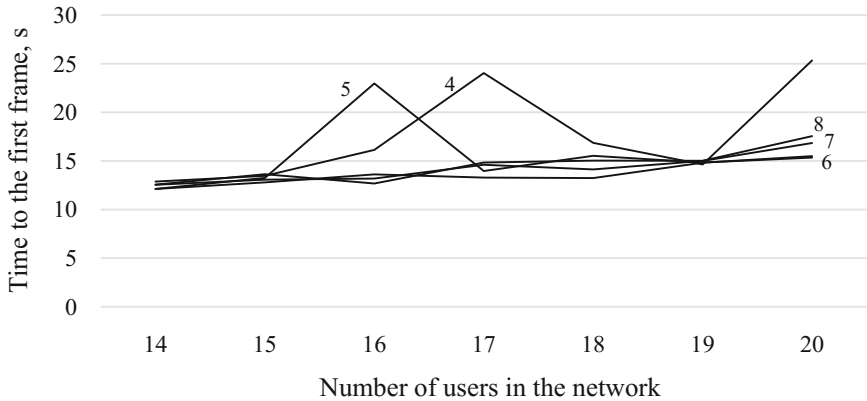
Fig. 6. Time to the first frame. Each line indicates the number of concurrent connections between users.

The time to the first frame can be reduced by finding a user that can accept a new connection faster. In the current version of the proposed system, each new user receives a list of other users in the swarm when it registers with the tracker-node. In a future version, each node might periodically report to the tracker node the number of free connection slots it has. The tracker then could include this information with a list of current users in the swarm that is sent to all new users.

6 Conclusions

This work evaluated a multimedia distribution system for use in urban rail environments. The proposed system uses Communications-Based Train Control radio link's unused bandwidth to deliver the multimedia stream to the train-car, where it is subsequently distributed to the train passengers using the peer-to-peer technologies.

The proposed system was tested using computer network emulation techniques. The results indicate that up to 14 users can concurrently receive the same high-definition (HD 720p) multimedia stream. The number of concurrent users can be increased to 18 by reducing the number of connections that each user maintains to other users from 10 to 4. The results with more than 18 concurrent users indicate that the user's experience, as measured by the playback index and the amount of discarded video data, falls sharply as the number of concurrent users increases further.

The evaluation results show that the proposed system can be used only in situations where the number of train passengers in a single train-car is limited. The main cause being that each user acts as a source of interference to other users during data transmissions. Future work is needed to improve the maximum number of users supported by the proposed system. Nevertheless, it shall be noted that the total number of train passengers that can use the system can be increased by deploying multiple instances of the system, e.g. one in each train-car.

In future work, the proposed system will use multiple non-overlapping Wi-Fi channels to deliver data between the users, reducing the radio interference between the users. Furthermore, advanced versions of data skipping and connections establishment algorithms will be researched to further improve the user experience of the system.

References

1. Ericsson Mobility Report - Western Europe. Ericsson (2016)
2. Cisco VNI Mobile Forecast (2015 – 2020). Cisco (2016)
3. Strategic Analysis of Communications-Based Train Control Systems in the Western European Urban Rail Market. Frost & Sullivan (2013)
4. Sector Overview and Competitiveness Survey of the Railway Supply Industry. European Commission (2012)
5. IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements (2004)
6. IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (2016)
7. Farooq, J., Soler, J.: Radio communication for communications-based train control (CBTC): a tutorial and survey. IEEE Commun. Surv. Tutorials 1 (2017)
8. Wi-Fi Peer-to-Peer (P2P) Technical Specification Version 1.7. Wi-Fi Alliance (2016)
9. Bakker, A., Petrocco, R., Grishchenko, V.: Peer-to-peer streaming peer protocol (PPSPP). IETF RFC7574 (2015)
10. Chandra, R., Karanth, S., Moscibroda, T., Navda, V., Padhye, J., Ramjee, R., Ravindranath, L.: DirCast: a practical and efficient Wi-Fi multicast system. In: 2009 17th IEEE International Conference on Network Protocols, pp. 161–170. IEEE (2009)
11. Kostuch, A., Gierłowski, K., Wozniak, J.: Performance analysis of multicast video streaming in IEEE 802.11 b/g/n testbed environment. In: Wozniak, J., Konorski, J., Katulski, R., Pach, Andrzej R. (eds.) WMNC 2009. IAICT, vol. 308, pp. 92–105. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03841-9_9
12. Chiao, H., Chang, S., Li, K., Kuo, Y., Tseng, M.: WiFi multicast streaming using AL-FEC inside the trains of high-speed rails (2010)
13. Camps-Mur, D., Garcia-Saavedra, A., Serrano, P.: Device-to-device communications with Wi-Fi Direct: overview and experimentation. IEEE Wirel. Commun. 20, 96–104 (2013)
14. Pyattaev, A., Galinina, O., Johnsson, K., Surak, A., Florea, R., Andreev, S., Koucheryavy, Y.: Network-assisted D2D over WiFi direct. In: Mumtaz, S., Rodriguez, J. (eds.) Smart Device to Smart Device Communication, pp. 165–218. Springer, Cham (2014). doi:10.1007/978-3-319-04963-2_7
15. Zhang, X., Hassanein, H.: A survey of peer-to-peer live video streaming schemes - an algorithmic perspective. Comput. Netw. 56, 3548–3579 (2012)
16. Osmani, F., Grishchenko, V., Jimenez, R., Knutsson, B.: Swift: the missing link between peer-to-peer and information-centric networks. In: Proceedings of Work. P2P Dependability, pp. 1–6 (2012)
17. Petrocco, R., Pouwelse, J., Epema, D.H.J.: Performance analysis of the Libswift P2P streaming protocol. In: 2012 IEEE 12th International Conference Peer-to-Peer Computing P2P 2012, Pp. 103–114 (2012)
18. Vlavianos, A., Iliofotou, M., Faloutsos, M.: BiToS: enhancing BitTorrent for supporting streaming applications. In: Proceedings - IEEE INFOCOM (2006)

19. Conti, M., Delmastro, F., Minutiello, G., Paris, R.: Experimenting opportunistic networks with WiFi Direct. In: Proceedings of IFIP Wireless Days (2013)
20. Ahrenholz, J.: Comparison of CORE network emulation platforms. In: Proceedings - IEEE Military Communications Conference MILCOM, pp. 166–171 (2010)
21. Extendable Mobile Ad-hoc Network Emulator (EMANE). https://www.nrl.navy.mil/itd/ncs/products/emane
22. Bankoski, J., Wilkins, P., Xu, Y.: Technical overview of VP8, an open source video codec for the web. In: Proceedings - IEEE International Conference Multimedia Expo (2011)